

## Application No. 09/173,858

1 6. (original) The interface of claim 1, wherein the machine readable specification  
2 includes a document compliant with a definition of an interface document including logical  
3 structures for storing an identifier of the interface, and for storing at least one of specifications  
4 and references to specifications of a set of one or more transactions supported by the interface.

1 7. (original) The interface of claim 6, wherein the machine readable specification  
2 includes a reference to a specification of a particular transaction, and the specification of the  
3 particular transaction includes a document including logical structures for storing at least one of  
4 definitions and references to definitions of input and output documents for the particular  
5 transaction.

B 1 8. (original) The interface of claim 1, wherein the storage units comprise parsed  
2 data.

1 9. (original) The interface of claim 8, wherein the parsed data in at least one of  
2 the input and output documents comprises:  
3 character data encoding text characters in the one of the input and output documents,  
4 and  
5 markup data identifying sets of storage units according to the logical structure of the  
6 one of the input and output documents.

1 10. (original) The interface of claim 9, wherein at least one of the sets of storage  
2 units encodes a plurality of text characters providing a natural language word.

1 11. (original) The interface of claim 8, wherein the interpretation information for at  
2 least one of the sets of storage units identified by a particular logical structure of at least one of  
3 the input and output documents, encodes respective definitions for sets of parsed characters.

## Application No. 09/173,858

1 12. (original) The interface of claim 8, wherein the storage units comprise unparsed  
2 data.

1 13. (original) The interface of claim 1, including a repository stored in memory  
2 accessible by at least one node in the network of document types for use in a plurality of  
3 transactions, and wherein the definition of one of the input and output documents includes a  
4 reference to a document type in the repository.

1 14. (original) The method of claim 13, wherein the repository of document types  
2 includes a document type for identifying participant processes in the network.

1 15. (original) The interface of claim 1, wherein the definitions of the input and  
2 output documents comprise document type definitions compliant with a standard Extensible  
3 Markup Language XML.

1 16. (original) The interface of claim 1, wherein the machine readable data structure  
2 including interpretation information comprises a document organized according to a document  
type definition compliant with a standard Extensible Markup Language XML.

1 61. (original) A method for programming a commercial transaction in a network,  
2 comprising  
3 defining a machine readable definition of an input document for a node in the network  
4 including resources to execute a process in the transaction, and a machine readable definition of  
5 an output document for the node, the definitions of the input and output documents comprising  
6 respective descriptions of sets of storage units and logical structures for the sets of storage  
7 units; and  
8 providing interpretation information for the logical structures to the node.

## Application No. 09/173,858

1           62. (original) The method of claim 61, wherein the interpretation information  
2 includes data type specifications for at least one logical structure in the definitions of the input  
3 and output documents.

1           63. (original) The method of claim 61, wherein the interpretation information  
2 includes at least one data structure mapping predefined sets of storage units for a particular  
3 logical structure in the definitions of the input and output documents, to respective entries in a  
4 list.

B 1           64. (original) The method of claim 61, the step of providing interpretation  
2 information includes providing a repository in memory accessible by at least one node in the  
3 network storing a library of logical structures, and interpretation information for logic  
4 structures.

1           65. (original) The method of claim 61, including defining a machine readable  
2 specification of an interface including a document compliant with a definition of an interface  
3 document including logical structures for storing an identifier of a particular transaction, and at  
4 least one of the definitions and references to the definitions of the input and output document.

1           66. (original) The method of claim 61, wherein the storage units comprise parsed  
2 data.

1           67. (original) The method of claim 66, wherein the parsed data in at least one of  
2 the input and output documents comprises:  
3 character data encoding text characters in the one of the input and output documents,  
4 and  
5 markup data identifying sets of storage units according to the logical structure of the  
6 one of the input and output documents.

Application No. 09/173,858

1 68. (original) The method of claim 67, wherein at least one of the sets of storage  
2 units encodes a plurality of text characters providing a natural language word.

1 69. (original) The method of claim 67, wherein the interpretation information for at  
2 least one of the sets of storage units identified by a particular logical structure of at least one of  
3 the input and output documents, encodes respective definitions for sets of parsed characters.

1 70. (original) The method of claim 66, wherein the storage units comprise unparsed  
2 data.

h 1 71. (original) The method of claim 61, wherein the definitions of the input and  
2 output documents comprise document type definitions compliant with a standard Extensible  
3 Markup Language XML.

1 72. (original) The method of claim 61, including:  
2 providing a parser to generate event signals in response to logical structures in the  
3 definition of the input document; and  
4 providing event listener programs which respond to the event signals to execute the  
5 process.